Volume 4, Issue 9, September 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

New Approach for 2D Image Segmentation

Fahad Alraddady
Computer Engineering Department, College of Computers and Information Technology,
Taif University, Taif, Saudi Arabia

Abstract—This paper presents a novel segmentation approach based on combining edge detection and region growing methods. The proposed algorithm starts by applying edge detection method to given images. Then, the region growing is selected a pixel on boundaries as an initial seed. The seed is grown by merging neighbouring pixels whose properties are most similar to the premerged region. These new pixels as a new seed pixel to continue the above process until no more pixels that satisfy the condition can be included. The region growing is used to obtain all closed edges in the image. The image is partitioned into sub-images according to the closed edges, where each sub-image contains segments, again detected using region growing, by eliminating all pixels contained in or on the sub-image. The results of the conducted experiments show that the efficiency of the proposed method in preserving the regions homogeneity and its robustness in segmenting noisy images is better than other region-based methods

Keywords— image segmentation, seed region growing (SRG), edge detection.

I. INTRODUCTION

The general segmentation problem involves the partitioning of a given 2D image into a number of homogeneous segments or spatially connected groups of pixels. Several techniques have been proposed to deal with the 2D image segmentation problem [1]. Among them edge and region growing based segmentation techniques appear stable and give accurate results.

Region and Edge-based segmentation [2-6] are the location of pixels in the image that correspond to the boundaries of the objects seen in the image. As boundary of a region or an object then it is closed and that the number of objects of interest is equal to the number of boundaries in an image. An edge or a linear feature is manifested as an abrupt change or a discontinuity in digital number of pixels along a certain direction in an image. It is calculated by using gradient or first and second order derivatives. Sobel [8], Canny [9], Prewitt and Roberts [10-11] are the famous edge detection methods.

Region-based techniques: take one or more pixels, called seeds, and grow the regions around them based upon a certain homogeneity criteria. If the adjoining pixels are similar to the seed, they are merged with them within a single region. The process continues until all the pixels in the image are assigned to one or more regions. property such as color, intensity and/or texture. To start a region growing initial position known as “seed” must be selected based on certain criteria. Seed Selection: In previous work [2,4,6-7] automatic seed is selected based on the following three criteria:

1. The seed pixel must have high similarity to its neighbors.
2. For an expected region, at least one seed must be generated in order to produce the region.
3. Seeds for different regions must be disconnected

The major problems of segmentation techniques that are based on region growing are accuracy of the segmentation and efficiency in terms of speed of region growing around the pixels. Also, seeded region growing (SRG) based methods are controlled by a number of initial seeds [12].

To alleviate these problems, the present paper introduces a new idea to combine edge-based and SRG method. A fast image segmentation method is presented to avoid the limitation of region growing and edge-based methods when they are combined together. The proposed method works by applying SRG on the edge pixels instead of image pixels as is usually the case in region growing techniques. The proposed algorithm begins by detecting image edges by using the Canny detector [9], where it is more stable and reliable edge detector. Most segmentation methods that combine edge detection with region growing have a simple control structure, and performs well in relatively simple images, these methods do not require any extra data beyond the position of the edge pixels. A post edge detection process detects holes and fills them with unit-width pixels. A procedure based on SRG for subdividing an image into sub-images is presented. The final procedure is to identify and extract all the segments from sub-images.

This paper is organized as follows: Section 2 describes the proposed method. The experimental results are presented in Section 3.Finally, conclusions are discussed in Section 4.

II. THE ALGORITHM

The In this section, we present an overview of the proposed algorithm and discuss its advantages. The first stage is to detect the edges for the image using a stable and reliable edge detector as given in [9]. Holes within object boundaries may result from extremely low or practically non-existent edge contrast. To close a hole in an edge, we developed a filling procedure to fill edge holes. Since we call the distance between a given two adjacent pixels a hole, if the distance between them is greater than a threshold (also smaller than another threshold), intermediate points are generated along the straight line between these pixels. Finally, a thinning procedure is used to remove spurious or unwanted generated points to ensure that the edges are of 1-pixel width.

The next stage is to subdivide the image. Although SRG techniques can perform this operation, they are computationally expensive due to the seed growing process acting on all pixels and the reliance on initial seed identification. A procedure based on SRG method is applied to edges pixels for subdividing an image into sub-images where each sub-image contains a segment and external pixels. Each sub-image is fed to the segment extraction algorithm, which implies a method for extracting a segment by applying the SRG technique on the external pixels. Then we use the transform process to convert the segment pixels in each sub-image to the corresponding pixels in the input image. The final stage is to extract the segments. For each sub-image we again use SRG to remove the external pixels and leave the required segment information.

The proposed method overcomes the computational limitation of SRG by reducing seed growing process. Furthermore, seeds automatically determined from the pixels on the left and right side of the boundary (edge) as shown in Fig.(3). This results in a fast, robust and stable algorithm for large image segmentation whose computational complexity is simplified. The problem of undesirable over segmentation produced by the Watershed algorithm applied to raw data images is avoided by extracting regions whose boundaries contain holes. Also, the edge maps we obtained have no broken lines on the entire image and the final edge detection result is one closed boundary per actual region in the image.

Our algorithm consists of the following steps:

- Edge detection.
- Image subdividing
- Segment extraction

In the following subsections, we will discuss these steps in more details.

A. EDGE DETECTION

Edge pixles are detected using the optimum algorithm proposed by Canny in [9]. The image is pre-processed to remove as much noise as possible using Gaussian derivatives. This process is necessary because the candidate edge pixels may contain wide ridges around the local maxima. Nonmaximum suppression thins such ridges to produce 1-pixel wide edges. The output of nonmaximum suppression still contains the local maxima created by noise. Hysteresis thresholding is used to get rid of these noises, which find chains of connected edge maxima, or connected contours. Due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis approach uses two thresholds, a high and a low (T1, T2) to reduce the probability of false contours. Any pixel in the image that has a value greater than T1 is set to either 1 or 255 (i.e. it is identified as an edge), and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value above T2 are also selected as edge pixels. If the value of any pixel lies between T1 and T2 and doesn't connected to edge pixels, it is set to zero (it is identified as a nonedge). Also, T2 is set to 0 if the value of a pixel is smaller than T2.

B. IMAGE SUBDIVISION

After detecting the closed edges (any continues edges that represent the boundaries of a region), image pixels are labeled as either edge or non-edge. All edge points are placed in an array E, and non-edge points are placed in an array P. In this section we present a procedure based on SRG that operates on the edge pixels to partition the image into a number of sub-images. We describe the algorithm in the following three subsections:

1. SRG METHOD

The simple RG technique consists in merging neighboring pixels Px of the pixel Py, inside the region, according to $|I(P_x) - I(P_y)| \leq T$, where T is a fixed threshold and $I(\bullet)$ is the pixel intensity value. This technique has two problems, 1) the choice of the threshold and 2) this technique can lead to a chaining effect especially for image with pixel intensity changing gradually. The second problem can be solved by using the homogeneity test $f(I_{i,j})=| I_{i,j} -RA| \leq T$, where RA is the region average (the summation of pixels intensities over the number of pixels inside the region).

This technique used different fixed threshold for each tissue in the image. The fixed threshold cross the function $f(I_{i,j})$ in two points ($a < RA$ and $b > RA$ see Fig.1) with the same distant from RA, if T which is a linear function is not correct and small then pixels inside the region will be described as outside, and if it big some pixels outside the region will be added to the region specially when the tissue has weak boundaries. This process is repeated until no more pixels are assigned to the region [13]. Since we only perform the seed growing on edge pixels, the amount of data needed to be processed is much reduced, resulting in increased speed. SRG algorithm iteratively merge similar pixels into sets or merge sub-regions into larger regions in 3 main steps: (1) choice of the seed pixels; (2) neighborhood analysis according to a similarity rule, and (3) grow the seed regions by including adjacent pixels that satisfy the similarity rule. The steps (1) and (2) are repeated until there are no more adjacent pixels to be included in a seed region. In the next section, we will solve the choice of threshold problem. The pixels neighbors in 2D are 8 pixels as shown in Fig. 2.

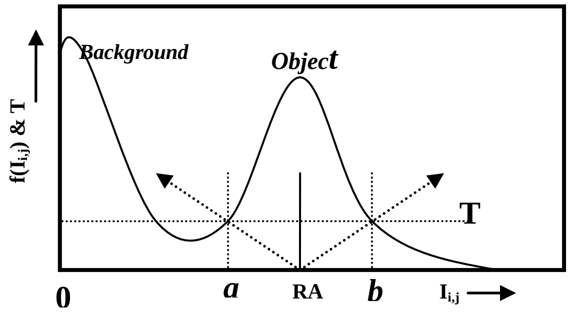


Fig. 1. The function *f(Ii,j)* with fixed threshold

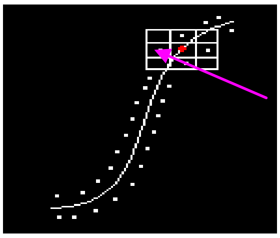
$I_{i-1,j-1}$	I_{i-1j}	$I_{i-1,j}$
$I_{i,-1,j}$	RA	$I_{i,j-1}$
I_{i-1j-1}	I_{i+1j}	$I_{i-1,,j+1}$

Fig. 2: Pixel and its neighbours for 2D.

2. FINDING A CLOSED EDGE

For each edge pixel in the array E, four arbitrary adjacent pixels are selected from the array P by applying a 3×3 mask. This can be done by making a mask on the edge. Although the result is not affected by increasing the size of the mask, the processing time increases linearly with the size of the mask. The seed pixels are chosen on the sides and along the edges, as illustrated in Fig. 3.

The four pixels adjacent to an edge pixel are placed in an array A. The SRG algorithm is now applied to the pixels in A to find a closed edge. An initial seed is an arbitrary non-edge pixel in A. The seed grows on one side from boundary (non-edge pixels), while each non-edge pixel is connected to an edge pixel (i.e. it grows on the non-edge pixels that are nearest to edge pixels on one side from boundary) and the resultant edge pixels are inserted into an array, Lbl. The seed grows continuously until a closed edge has been obtained. The edge pixels of Lbl1 represent the edge pixels of edge E1. The process is repeated for each closed edge Ei in E until A becomes empty.



Mask on an edge

Fig. 3: Two sets of seeds are located at both sides of an edge (the pixels on left and right sides of the boundary).

3. SUBDIVIDING AN IMAGE INTO SUB-IMAGES

The edge arrays E_i are sorted in increasing number of pixels. Each array is then sorted in increasing x or y values for processing the segment with least number of pixels first. For each Ei, pixels with the minimum and maximum x and y values are detected. A small user defined integer threshold ε is added to guarantee a region is extracted without losing any edge points. Increasing the threshold increases the size of the sub-image and hence the time to process but will not effect the resulting segmentation. A sub-image contained in the Cartesian bounding box: (xmax+ ε ,y) ; (xmin- ε ,y) ; (x, ymin $-\varepsilon$) ; (x, ymax $+\varepsilon$) is extracted and stored. This is repeated until all sub-images for each closed edge have been removed, as shown in Fig. 4.

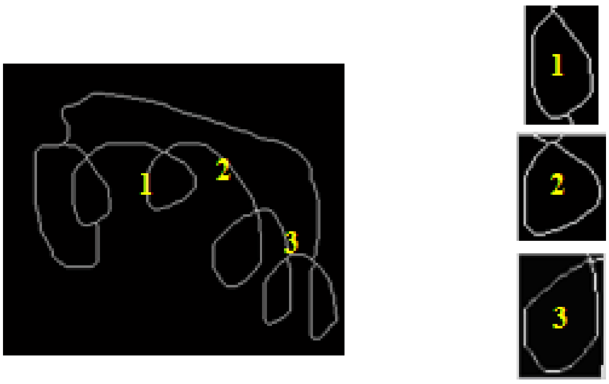


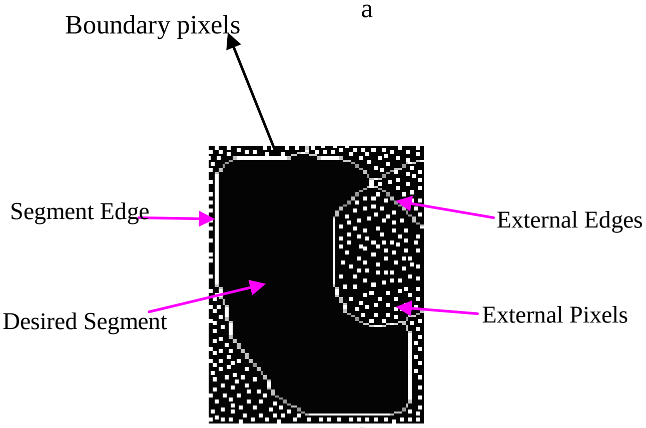
Fig. 4: The image is partitioned into sub-images.
For instance the sub-images 1, 2, and 3 are obtained from the image.

C. SEGMENTS EXTRACTION

At this stage, the image is partitioned into sub-images, where each sub-image is bounded by a closed edge. In this section, we present an algorithm based on SRG for obtaining a segment from a sub-image. The algorithm is only applied to external pixels, i.e. pixels that lie outside a segment (see Fig. 5(b)).

2	2	2	2	2	2	2	2	2	2	2	2	2
2	0	0	0	1	1	1	0	0	0	0	1	2
2	0	0	1	0	0	0	1	0	0	1	0	2
2	0	1	0	0	0	0	0	1	1	1	0	2
2	0	1	0	0	0	0	0	0	1	0	1	2
2	0	1	0	0	0	0	0	0	1	0	0	2
2	0	0	1	0	0	0	0	1	0	0	0	2
2	0	0	0	1	0	0	0	1	0	0	0	2
2	0	0	0	0	1	0	0	1	0	0	0	2
2	0	0	0	0	0	1	1	0	0	0	0	2
2	0	0	0	0	0	1	0	0	0	0	0	2
2	0	0	0	0	0	1	0	0	0	0	0	2
2	2	2	2	2	2	2	2	2	2	2	2	2

a



b

Fig. 5: (a) The binary sub-image surrounded by labeled boundary,
(b) a segment includes external edges, segment edges, and external pixels.

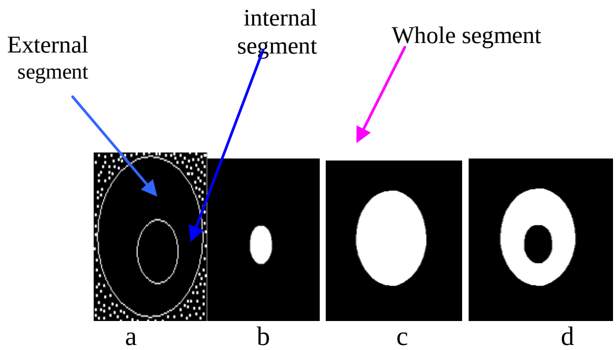


Fig.6: (a) a segment with internal segment, (b) the stage of extracting the small segment,
(c) and (d) the stages of extracting the large segment.

The proposed algorithm consists of three steps: labeling the background of the sub-image, segment extraction, and converting the pixels in the sub-image to corresponding pixels in the original image. These steps are:

Step 1: Labeling the background in the sub-image:

Starts with labeling pixels of the boundary of a sub-image, a pixels takes 2 if it represents a boundary pixel. All pixels of the top, bottom, left, and right lines (called boundary pixels) of each sub-image are labeled with 2. To label all pixels in the background, SRG is applied to non-edge pixels in the same way as the edge pixels, where the initial seeds RA are selected automatically from the boundary pixels, as shown in Fig. 5(a).

Step 2: Converting the pixels in the segment to corresponding pixels in the image:

The remaining unlabeled pixels will represent the desired segment. After obtaining the desired segment from sub-images, we want to convert the pixels in the sub-image into the corresponding pixels in the original image. This can be done by translating the coordinate's pixels from sub-images to original image using the transformations $xorg=x+xmin-\varepsilon$, $yorg=y+ymin-\varepsilon$, where (x,y) and $(xorg,yorg)$ are the pixels coordinates in sub-images, and original image, respectively.

Step 3: Segment modification:

The goal of this step is to obtain the desired segment from a whole segment i.e. a segment that includes an internal segment. In this step, after translating the whole segment from the sub-image space to entire image space, the whole segment is tested, if it contains an internal segment, then it is easy to extract the internal segment by comparing the internal segment pixels with the extracted one. Note that, the internal segment is extracted before the external one because it has a smaller number of pixels than the external segment. The residual segment (whole segment minus the internal segment) represents the desired segment in the original image, as shown in Fig. 6(b)-6(d).

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

Here, we present a set of experimental results to show the effectiveness of the proposed technique when applied on a range of images. The images that have been used here include whole segments to illustrate the performance of the proposed algorithm.

Since most hybrid techniques are based on SRG [13] or Watershed methods [14], thus we compare the proposed image with SRG method and Watershed method based on estimating the time for each method and segmentation accuracy when applied to different test images: synthetic as shown in Fig.7(a,d), and real world, as shown in Fig. 8(a), Fig. 9(a), and Fig. 10(a) , Fig. 11(a).

The presented examples also serve to demonstrate the steps of the proposed algorithm. As described, the algorithm works automatically without the need to specify initial seeds. The threshold T is between values, where η_{min} , η_{max} , and ε are selected in all experimental examples to be 2, 5, and 3, respectively.

A. Segmentation Accuracy

To assess the accuracy of the segmentation methods two simulated images of known segments are contaminated with Gaussian noise and then processed using SRG, Watershed and the proposed methods, The first simulation image contains two regions (classes) and the second image contains four regions. Table 2 presents the segmentation accuracy scores for three different noise levels (1%, 3% and 5%) as in Figs. 7(a, d), Fig.7(b, e), Fig. 7(c, f) respectively.

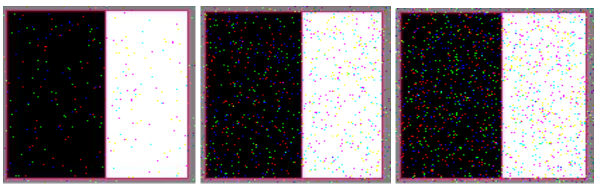
The overall segmentation accuracy is computed by dividing the total number of correctly classified pixels over the total number of pixels [15]. From Table 2 we see that, as the level of noise increases, the numbers of misclassified pixels of all three algorithms increase as expected, but in all cases SRG and the proposed method achieved much better performance than the Watershed method. Table 2 also shows that the proposed method is as least as good as SRG and is better as the level of noise increases. These results suggest that the proposed method is worthy of further consideration.

B. Time Overhead

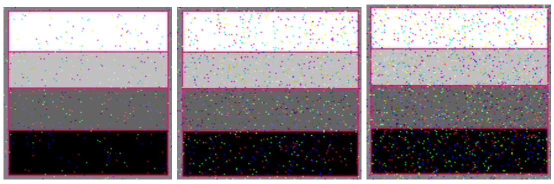
Here, the complexity of the processing time depends on the number of edge pixels and not the number of image pixels. This will extremely reduce the time which is required for growing the regions around the pixels [16-17]. Four real world images are used, "sim2", "flower", "blood", and "house" to show the efficiency of using the proposed method, where these images may be difficult to segment due to including hole edges. The results of the first stage of the algorithm that applied to the four test images are presented in Figs. 7(b), 8(b), 9(b), 10(b) and 11(b) respectively, where the goal of the first stage is to detect the edges and modify these edges through closing holes and the edge thinning for the image. Figs. 8(c), 9(c), 10(c) and 11(c) illustrate some sub-images that were identified by the image subdividing procedure. SRG is used to obtain all closed edges in the image, where the image is partitioned into sub-images according to these closed edges. Figs 8(d), 9(d), 10(d) and 11(d) depict the segments, which are extracted from the sub-images by performing the region extracting procedure, since SRG is used again to remove the external pixels to leave the required segment information. Figs. 8(e), 9(e), 10(e) and 11(e) display the resultant segments after applying segments extraction procedure.

We compare the performance of the traditional SRG, Watershed and the proposed method. The comparison is based on estimating the time for each method when applied to the four test images. Table 3 shows typical execution times of the traditional SRG, Watershed and the proposed method. The running time is dependent on the number of non-edge pixels, i.e. if the number of non-edge pixels is high, the execution time will be large. For instance, for "sim2" and "flower" images, the total number of non-edge pixels is 49967 and 42242, respectively, the executing time for SRG is 17.0470 and 15.5470, and Watershed is 26.3440 and 22.0780 respectively. Note, these times do not include the time required for manually selecting thresholds or initial region seeds. From this table, the proposed method is faster than other methods because our method only works on edge pixels.

Theoretically, assume the number of edge pixels is m and the number of nonedge plus edge pixels is n, and let f(n) represents the computation complexity of method that based on all pixels. Since the number of non-edge pixels is greater than the number of edge pixels, the relation $m \ll n \Rightarrow f(m) \ll f(n)$ is always true. This shows that the computation complexity of SRG techniques is reduced when we work only with the edges pixels of an image as in the case of the proposed method.



A B C



D E F

Fig. 7: Two synthetic images with three different noise levels (1%, 3% and 5%). (a), (b), and (c) synthetic1 with different noise levels 1%, 3%, and 5% respectively. (d), (e), and (f) synthetic1 with different noise levels 1%, 3%, and 5% respectively.

TABLE II. Segmentation accuracy (%) of three methods on synthetic images.

Methods	Simulated 1			Simulated 2		
	1%	3%	5%	1%	3%	5%
Watershed	92.33	82.00	72.19	90.09	77.99	67.62
SRG	97.09	91.42	83.64	96.53	88.86	82.83
Proposed-method	97.09	91.46	83.75	96.66	89.89	83.92

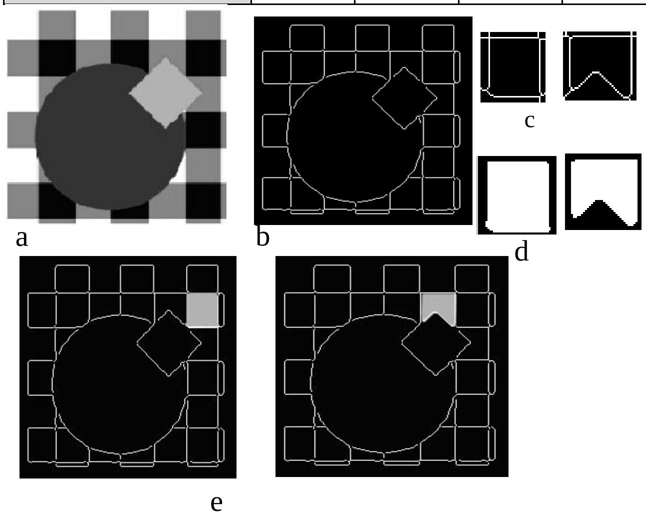


Fig. 8: Segmentation results of "sim2", (a) original image, (b) image after closing holes and thinning edge, (c)-(d) some sub-images of the image after performing subdividing procedure, and (e) the segment is extracted from a sub-image.

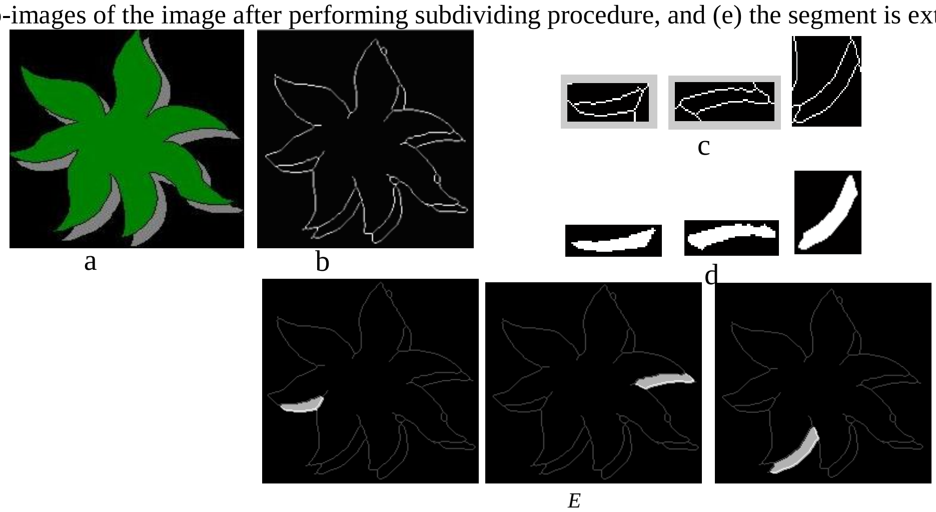
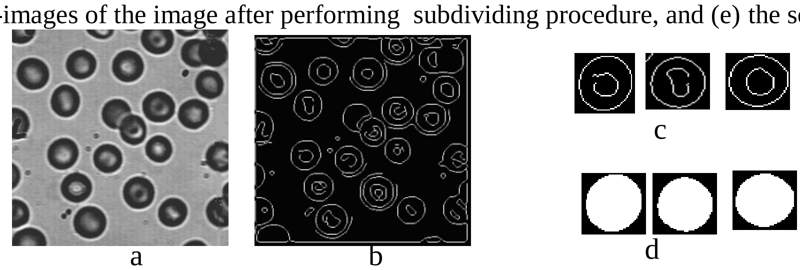
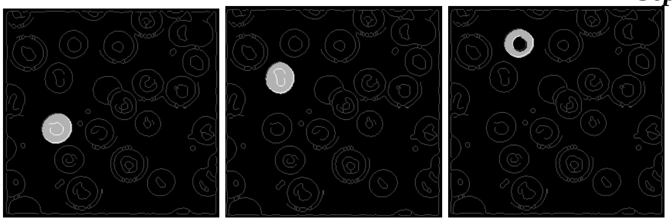


Fig. 9: Segmentation results of "flower", (a) original image, (b) image after closing holes and thinning edge, (c)-(d) some sub-images of the image after performing subdividing procedure, and (e) the segment is extracted from a sub-image.





e

Fig. 10: Segmentation results of "blood", (a) original image, (b) image after closing holes and thinning edge. (c)-(d) some sub-images after performing subdividing procedure, and (e) the segment is extracted from a sub-image.

Fig. 11: Segmentation results of "house":(a) original image, (b) image after closing holes and thinning edge, (c)-(d) some sub-images after performing subdividing procedure, and (e) the segment is extracted from a sub-image.

TABLE III. Execution times of the SRG, Watershed, and the proposed method

Image	Size	Number		Time (second)		
		Edge	Non-edge	SRG	Watershed	Proposed method
sim2	221×236 V. VI.	2189	49967	17.0470	26.3440	9.9744
flower	209 ×209 VII.	1439	42242	15.5470	22.0780	5.1728
blood	283 ×274	6894	70648	22.5780	38.7970	14.371
house	308 ×278	4344	81280	27.8280	43.8750	13.872

IV. CONCLUSION

In this paper, we proposed a hybrid image segmentation method that combines edge-based and region-based methods. It makes the segmentation results robust, since the seed regions are selected automatically, also gives a substantial speedup compared to the traditional SRG and Watershed methods implementation for many images.

By comparing the proposed method with traditional SRG and Watershed method, we found that the proposed hybrid image segmentation technique performs better for several reasons: Since processing time depends on the complexity of the image, it can be reduced by reducing the data volume of the image from total pixels to sub-regions. So, the time overhead of data volume is reduced by subdividing the image into sub-images with only growing on the edge pixels, and this speeds up the algorithm. Also accurate segmentation results have been obtained and improved for many images with hole edges without controlling by the initial seeds, since they are automatically selected.

Future research is directed toward the extension to a 3-D version of the algorithm and its application to segmentation of moving 3-D images. The quality improvements of edge thinning and closing of 2-D and 3-D images are still open research topics.

REFERENCES

[1] N. Pal, and S. Pal, “A review on image segmentation techniques,” Pattern Recognition. vol. 26, pp. 1277–1294, 1993.

[2] R.B. Ohlander, "Analysis of natural scenes," PhD Thesis, Carnegie Institute of Technology, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1975.

[3] N. Otsu, "A threshold selection method from grey level histograms," IEEE Transactions on Systems, Man and Cybernetics, vol.. 8, pp. 62-66, 1978.

[4] L.Padma Suresh and MMary Synthuja Jain. Image segmentation using seeded region growing. 2012.

[5] E.A.Zanaty, Sultan Aljahdali, " Semi-automatic Region Growing Method for Brain MRIs Segmentation", Accepted to publish in Society for Computers and Their Applications (ISCA) journal, 2014.

[6] E.A.Zanaty, Ahmed S. Ghiduk, "A novel approach for medical image segmentation based on genetic and seed region growing algorithms", Journal of Computer Science and Information Systems ComSIS Vol. 10, No. 3, pp.1319-1342, 2013.

[7] M. Cheriet, J. N. Said and C. Y. Suen, "A recursive thresholding technique for image segmentation," IEEE Transactions on Image Processing, vol. 7, No. 6, pp. 918-920, 1998.

[8] D. Marr, and E. Hildreth, "Theory of edge detection," in Proc. R. Soc. Lond. B, No. 207, pp. 187–217, 1980.

[9] J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Machine. Intell., vol. PAMI-8, pp. 679–698, Nov. 1986.

[10] V. Torre, and T. Poggio, "On edge detection," IEEE Trans. Pattern Anal. Machine. Intell, vol. PAMI-8, pp. 147–163, Mar. 1986.

[11] O. Monga, R. Deriche, G. Malandain, and J. P. Cocquerez, "Recursive filtering and edg tracking: Two primary tools for 3D edge detection," Image Vis. Comput., vol. 9, pp. 203–214 Aug. 1991.

[12] R. Adams, and L. Bischof, "Seeded region growing," IEEE Trans. Pattern Anal. Machine Intell., vol. 16, pp. 641–647, 1994.

[13] J. Fan, G. Zeng, M. Body, M. Hacid, "Seeded region growing: an extensive and comparative study," Pattern Recognition Letters, vol. 26, pp.1139-1156, 2005.

[14] L. Vincent, P. Soille, "Watershed in digital space: an efficient algorithm based on immersion simulation," IEEE Trans. Pattern Anal. Mach. Intell, vol. 13, No. 6, pp. 583–598, 1991.

[15] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. Farag, T. Moriarty, " A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data," IEEE Trans Med Imaging, vol. 21, No. 3, pp. 193-199, 2002.

[16] S. A. Hijjatoleslami, and J. Kittler, "Region growing: A new approach," IEEE Trans. Image Processing, vol. 7, pp. 1079–1084, 1998.

[17] F. Jianping, K. Y. David, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," IEEE Trans. on image processing, vol. 10, No. 10, pp. 1454-1466, OCT. 2001.